

CLAIMS

I claim:

- 445 1) A method for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the method comprising
- a) eliminating programming comments from the first source code file;
 - b) eliminating programming comments from the second source code file;
 - 450 c) substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said first file;
 - d) substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in
 - 455 e) putting each remaining line of functional programming code of the first file into an array of text strings;
 - f) putting each remaining line of functional programming code of the second file into a second array of text strings; and
 - 460 g) finding all matches between text strings in said first array with text strings in said second array.
- 2) The method of claim 1) where finding all matches ignores the type case of the text.
- 3) A method for comparing two program source code files to help an expert
- 465 determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the method comprising
- a) eliminating functional programming lines from the first source code file, leaving comment lines;
 - 470 b) eliminating functional programming lines from the second source code file, leaving comment lines;

- c) substituting a single space character for sequences of whitespace characters in each remaining comment line in said first file;
 - d) substituting a single space character for sequences of whitespace characters in each remaining comment line in said second file;
 - e) putting each remaining comment line of the first file into an array of text strings;
 - f) putting each remaining comment line of the second file into a second array of text strings; and
 - g) finding all matches between text strings in said first array with text strings in said second array.
- 4) The method of claim 3) where finding all matches ignores the type case of the text.
- 5) A method for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the method comprising
- a) extracting all words between whitespace from each line of functional programming code in the first source code file to an array of text strings;
 - b) eliminating programming language keywords from said array of text strings;
 - c) extracting all words between whitespace from each line of functional programming code in the second source code file to a second array of text strings;
 - d) eliminating programming language keywords from said second array of text strings;
 - e) finding all matches between text strings in said first array with text strings in said second array.
- 6) The method of claim 5) where finding all matches ignores the type case of the text.

- 505 7) A method for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the method comprising
- a) extracting all words between whitespace from each line of functional programming code in the first source code file to an array of text strings;
 - b) eliminating programming language keywords from said array of text strings;
 - c) extracting all words between whitespace from each line of functional programming code in the second source code file to a second array of text strings;
 - d) eliminating programming language keywords from said second array of text strings;
 - e) finding all partial matches between text strings in said first array with text strings in said second array, where a partial match is where one string can be found in its entirety in as a second string but the strings are not identical.
- 520 8) The method of claim 7) where finding all partial matches ignores the type case of the text.
- 525 9) A method for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the method comprising
- a) eliminating programming comments from the first source code file;
 - b) eliminating programming comments from the second source code file;
 - c) substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said first file;
- 530

- d) substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said second file;
- e) putting each remaining line of functional programming code of the first file into an array of text strings;
- f) putting each remaining line of functional programming code of the second file into a second array of text strings; and
- g) finding sequences where the first word of each line in said first array matches the first word of each line in said second array.
- 10) The method of claim 9) where finding sequences where the first word of each line in said first array matches the first word of each line in said second array ignores the type case of the text.
- 11) A method for comparing two program source code files, comprising:
- a) extracting from each program source code file a first set of code elements and a second set of code elements;
- b) computing a first metric derived from comparing the first set of code elements for the first program source code file to the first set of code elements for the second program source code file;
- c) computing a second metric derived from comparing the second set of code elements for the first program source code file to the second set of code elements for the second program source code file;
- d) combining the first metric and the second metric to derive a combined metric, wherein the first and second sets of code elements are selected from the group consisting of complete words, selected partial words, selected source lines, selected comment lines and selected code sequences.
- 12) An apparatus for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the apparatus comprising
- A computer;

A source code matching program on said computer, wherein said source code matching program comprises:

- a) means for eliminating programming comments from the first source code file;
- b) means for eliminating programming comments from the second source code file;
- c) means for substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said first file;
- d) means for substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said second file;
- e) Putting each remaining line of functional programming code of the first file into an array of text strings;
- f) means for putting each remaining line of functional programming code of the second file into a second array of text strings; and
- g) means for finding all matches between text strings in said first array with text strings in said second array.

13) The apparatus of claim 11) where means for finding all matches ignores the type case of the text.

14) An apparatus for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the apparatus comprising

A computer;

A source code matching program on said computer, wherein said source code matching program comprises:

- a) means for eliminating functional programming lines from the first source code file, leaving comment lines;

b) means for eliminating functional programming lines from the second source code file, leaving comment lines;

c) means for substituting a single space character for sequences of whitespace characters in each remaining comment line in said first file;

d) means for substituting a single space character for sequences of whitespace characters in each remaining comment line in said second file;

e) means for putting each remaining comment line of the first file into an array of text strings;

f) means for putting each remaining comment line of the second file into a second array of text strings; and

g) means for finding all matches between text strings in said first array with text strings in said second array.

15) The apparatus of claim 14) where means for finding all matches ignores the type case of the text.

16) An apparatus for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the apparatus comprising

A computer;

A source code matching program on said computer, wherein said source code matching program comprises:

a) means for extracting all words between whitespace from each line of functional programming code in the first source code file to an array of text strings;

b) means for eliminating programming language keywords from said array of text strings;

c) means for extracting all words between whitespace from each line of functional programming code in the second source code file to a second array of text strings;

d) means for eliminating programming language keywords from said second array of text strings;

625 e) means for finding all matches between text strings in said first array with text strings in said second array.

17) The apparatus of claim 16) where means for finding all matches ignores the type case of the text.

18) An apparatus for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the apparatus comprising

630

A computer;

A source code matching program on said computer, wherein said source code matching program comprises:

635 a) means for extracting all words between whitespace from each line of functional programming code in the first source code file to an array of text strings;

b) means for eliminating programming language keywords from said array of text strings;

640 c) means for extracting all words between whitespace from each line of functional programming code in the second source code file to a second array of text strings;

d) means for eliminating programming language keywords from said second array of text strings;

645 e) means for finding all partial matches between text strings in said first array with text strings in said second array, where a partial match is where one string can be found in its entirety in as a second string but the strings are not identical.

19) The apparatus of claim 18) where means for finding all partial matches ignores the type case of the text.

650

20) An apparatus for comparing two program source code files to help an expert determine whether one file contains source code that has been

copied from the other file or whether both files contain code that has been copied from a third file, the apparatus comprising

655

A computer;

A source code matching program on said computer, wherein said source code matching program comprises:

a) means for eliminating programming comments from the first source code file;

660

b) means for eliminating programming comments from the second source code file;

c) means for substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said first file;

665

d) means for substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said second file;

e) means for putting each remaining line of functional programming code of the first file into an array of text strings;

670

f) means for putting each remaining line of functional programming code of the second file into a second array of text strings; and

g) means for finding sequences where the first word of each line in said first array matches the first word of each line in said second array.

675

21) The apparatus of claim 20) where means for finding sequences where the first word of each line in said first array matches the first word of each line in said second array ignores the type case of the text.

22) An apparatus for comparing two program source code files, comprising:

A computer;

680

A source code matching program on said computer, wherein said source code matching program comprises:

a) means for extracting from each program source code file a first set of code elements and a second set of code elements;

685 b) means for computing a first metric derived from comparing the first set of code elements for the first program source code file to the first set of code elements for the second program source code file;

c) means for computing a second metric derived from comparing the second set of code elements for the first program source code file to the second set of code elements for the second program source
690 code file;

d) means for combining the first metric and the second metric to derive a combined metric, wherein the first and second sets of code elements are selected from the group consisting of complete words, selected partial words, selected source lines, selected comment
695 lines and selected code sequences.